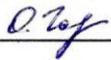


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Электронные вычислительные машины»

«СОГЛАСОВАНО»

Декан факультета ИЭ

 / Горбова О.Ю./  
«\_\_» \_\_\_\_\_ 2020г

«УТВЕРЖДАЮ»

Проректор РОНиМД

 / Корячко А.В.  
«\_\_» \_\_\_\_\_ 2020г



Заведующий кафедрой ЭВМ

 / Костров Б.В.  
«\_\_» \_\_\_\_\_ 2020г

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

### Б1.В.ДВ.05.01 «Объектно-ориентированное программирование»

Направление подготовки  
38.03.05 – «Бизнес-информатика»

Направленность (профиль) подготовки  
«Бизнес-информатика»

Уровень подготовки - бакалавриат

Квалификация выпускника – бакалавр

Формы обучения – заочная

Рязань 2020 г

## ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа составлена с учетом требований федерального государственного образовательного стандарта высшего образования по направлению подготовки (специальности) 38.03.05 – «Бизнес-информатика», утвержденного 11.08.2016 (приказ № 1002).

Разработчики

доцент кафедры ЭВМ Громов А.Ю.

\_\_\_\_\_/Громов А.Ю./  
(подпись)

Программа рассмотрена и одобрена на заседании кафедры ЭВМ «\_\_» \_\_\_\_\_ 2020г.,  
протокол № \_\_\_\_\_

Заведующий кафедрой

«Электронные вычислительные машины»

Костров Б.В.

\_\_\_\_\_/Костров Б.В./  
(подпись)

## 1 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Целью изучения дисциплины «Объектно-ориентированное программирование» является ознакомление студентов с принципами и подходами разработки прикладного и системного программного обеспечения (ПО) и информационных систем (ИС) с помощью современных систем программирования (СП) на уровне, который позволил бы квалифицированно разрабатывать программные продукты различной степени сложности, а также развить у слушателей курса практические навыки создания таких продуктов.

Обучение студентов по курсу «Объектно-ориентированное программирование» направлено на то, чтобы способствовать воспитанию у них стремления к постоянному повышению профессиональной компетентности, профессионального кругозора, умения ориентироваться в тенденциях и направлениях развития новых информационных технологий в области создания ПО.

### Перечень планируемых результатов обучения по дисциплине

Коды компетенции	Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
ПК-3	Выбор рациональных информационных систем и информационно-коммуникативных технологий решения для управления бизнесом	<u>Знать:</u> инструменты и методы проектирования и дизайна ИС, инструменты и методы верификации структуры программного кода, языки программирования и работы с базами данных, современные объектно-ориентированные языки программирования; <u>Уметь:</u> кодировать на языках программирования, верифицировать структуру программного кода; <u>Владеть:</u> методами анализа предметной области задачи. разработки структуры программного кода ИС, верификации структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС, устранени обнаруженных несоответствий.

### 2 Место дисциплины в структуре ОПОП

Дисциплина «Объектно-ориентированное программирование» относится к вариативной части блока Б1 (Б1.В.ДВ.05.01) профессионального цикла дисциплин основной образовательной программы подготовки бакалавров направления 38.03.05 «Бизнес-информатика».

Для изучения дисциплины обучаемый должен

*знать:*

- основы структурного программирования;
- базовые понятия программирования на языке высокого уровня;

*уметь:*

- разрабатывать алгоритмы;

– выполнять декомпозицию предметной области задачи;

*владеть:*

– навыками обобщения информации.

Данная дисциплина базируется на следующих дисциплинах учебного плана: «Информатика», «Программирование», «Математическая логика и теория алгоритмов».

Знания, полученные по освоению дисциплины, необходимы при изучении дисциплин: «Информационные технологии в бизнесе» и «Интеллектуальные системы».

### **3 Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся**

Общая трудоемкость дисциплины составляет 4 зачетных единицы (ЗЕ), или 144 часа.

Вид учебной работы	Всего часов	Семестр
		4
Общая трудоёмкость час.	144	144
	4	4
Контактная работа обучающихся с преподавателем (всего)	12,35	12,35
В том числе:		
Лекции	4	4
Лабораторные работы (ЛР)	4	4
Практические занятия (ПЗ)	2	2
Консультации	2	2
Иная контактная работа	0,35	0,35
Самостоятельная работа студентов (всего)	131,65	131,65
В том числе:		
Самостоятельные занятия	123	123
Экзамены и консультации	8,65	8,65
Консультации в семестре	-	-
Курсовой проект (работа)	-	-
Вид итогового контроля		экзамен

**4 Содержание дисциплины, структурированное по темам (разделам)  
с указанием отведенного на них количества академических часов  
и видов учебных занятий**

**4.1 Содержание дисциплины, структурированное по темам**

Раздел дисциплины	Содержание
1. Теоретические основы объектно-ориентированного программирования	<p>Основные этапы развития программирования. Структурная декомпозиция и модульное программирование. Основания и сущность объектно-ориентированного подхода. Языки ООП. Архитектура ООП-программ. Отличительные особенности ООП-подхода в программировании. Основные принципы объектно-ориентированного программирования. Объектная декомпозиция. Примеры. Понятие объекта. Объекты и сообщения. Состояние и поведение объектов. Понятие класса. Структура и организация определения класса. Примеры. Средства разработки классов. Наследование. Переопределение функций базового класса в производном классе. Инициализация объектов производных классов. Наследование конструкторов. Управление доступом к элементам класса в связи с наследованием. Особенности реализации наследования в некоторых языках ООП. Полиморфизм как средство разработки классов. Понятие о статическом и динамическом связывании. Простой полиморфизм. Примеры. Сложный полиморфизм и создание полиморфных объектов. Примеры.</p>
2. Принципы создания приложений Windows	<p>Особенности и основные средства ОС Windows, используемые при разработке приложений. Структура приложений Windows. Основные этапы разработки объектно-ориентированных приложений.</p> <p>Общая схема формальной верификации. Методы формальной верификации: дедуктивная верификация, проверка моделей, проверка эквивалентности.</p>

3. Особенности объектной модели языка Object Pascal	Особенности реализации полиморфизма в языке Object Pascal. Перегрузка процедур, функций и методов. Свойства в языке Object Pascal. Метаклассы в языке Object Pascal. Средства определения типов на этапе выполнения программ. Делегирование в языке Object Pascal. Библиотека стандартных классов в языке Object Pascal. Создание и обработка сообщений в языке Object Pascal.
4. Основы разработки приложений в управляемых вычислительных средах	<p>Понятия управляемого и неуправляемого программного кода. Управляемый код в языке Java. Управляемый код платформы Microsoft .NET.</p> <p>Классы в языке C#. Классы и пространство имен. Определение конструкторов и инициализация объектов в языке C#. Доступ к членам класса в языке C#. Создание и использование свойств в языке C#. Перегрузка операций в языке C#.</p>

#### 4.2 Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

Тема	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем						Самостоятельная работа обучающихся
		Всего	Лекции	Практические занятия	Лабораторные работы	Консультации	ИКР	
Тема 1. Теоретические основы объектно-ориентированного программирования.	38	6	2	2	2	-	-	32
Тема 2. Принципы создания приложений Windows.	35	5	2	-	2	1	-	30
Тема 3. Особенности объектной модели языка Object Pascal.	31	1	-	-	-	1	-	30
Тема 4. Основы разработки приложений в управляемых вычислительных средах.	31	-	-	-	-	-	-	31
Контроль	8,65	-	-	-	-	-	-	8,65
Промежуточный контроль	0,35	0,35	-	-	-	-	0,35	-
<b>Всего:</b>	<b>144</b>	<b>12,35</b>	<b>4</b>	<b>2</b>	<b>4</b>	<b>2</b>	<b>0,35</b>	<b>131,65</b>

### Виды практических и самостоятельных работ

Тема	Вид работы	Наименование и содержание работы	Трудо-емкость, часов	
Тема 1. Теоретические основы объектно-ориентированного программирования.	Практические работы	Знакомство со средой визуального программирования RAD Studio	2	
	Лабораторные работы	Принципы объектно-ориентированного проектирования программ в системе программирования RAD Studio	2	
	Самостоятельная работа		Изучение конспекта лекций	4
			Изучение методических указаний, подготовка к практическим занятиям	2
			Подготовка к выполнению и защите лабораторных работ	4
			Изучение теоретического материала по источникам	18
		Выполнение заданий контрольной работы	2	
		Подготовка к экзамену и консультации	2	
Тема 2. Принципы создания приложений Windows.	Лабораторные работы	Обработка списков строк и исключительных ситуаций	4	
	Самостоятельная работа		Изучение конспекта лекций	4
			Подготовка к выполнению и защите лабораторных работ	6
			Изучение теоретического материала по источникам	16
			Выполнение заданий контрольной работы	2
		Подготовка к экзамену и консультации	2	
Тема 3. Особенности объектной модели языка Object Pascal.	Самостоятельная работа	Изучение теоретического материала по источникам	26	
		Выполнение заданий контрольной работы	2	
		Подготовка к экзамену и консультации	2	
Тема 4. Основы разработки приложений в управляемых вычислительных средах.	Самостоятельная работа	Изучение теоретического материала по источникам	27	
		Выполнение заданий контрольной работы	2	
		Подготовка к экзамену и консультации	2	

### 5 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

1. Конкин Ю.В. . Объектно-ориентированное программирование. Методические указания к лабораторной работе. Часть 8. № 4867. РГРТУ, Рязань, 2014.
2. Конкин Ю.В. . Объектно-ориентированное программирование. Методические указания к лабораторной работе. Часть 9. № 4905. РГРТУ, Рязань, 2015.
3. Гринченко Н.Н., Конкин Ю.В. Разработка моделей информационных систем на языке UML: учеб. пособие / РГРТУ. - Рязань, 2015. - 48с.

## **6 Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине**

Фонд оценочных средств приведен в Приложении А.

## **7 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

### **Основная учебная литература**

1. Вендров А.М. Современные технологии создания программного обеспечения [Электронный ресурс] / «CitForum». URL: <http://citforum.ru/programming/application/program/> (дата обращения: 01.12.2014).
2. Липаев В.В. Программная инженерия сложных заказных программных продуктов [Электронный ресурс]: учебное пособие/ Липаев В.В.- Электрон. текстовые данные.- М.: МАКС Пресс, 2014.- 309 с.- Режим доступа: <http://www.iprbookshop.ru/27297>.- ЭБС "IPRbooks" (дата обращения: 21.06.2016).
3. Алексеев В.Е. Графы и алгоритмы. Структуры данных. Модели вычислений [Электронный ресурс]/ Алексеев В.Е., Таланов В.А.- Электрон. текстовые данные.- М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.- 153 с.- Режим доступа: <http://www.iprbookshop.ru/52186>.- ЭБС "IPRbooks" (дата обращения: 21.06.2016).
4. Казанский А.А. Объектно-ориентированное программирование на языке Microsoft Visual C# в среде разработки Microsoft Visual Studio 2008 и .NET Framework. 4.3 [Электронный ресурс]: учебное пособие и практикум/ Казанский А.А.- Электрон. текстовые данные.- М.: Московский государственный строительный университет, ЭБС АСВ, 2011.- 180 с.- Режим доступа: <http://www.iprbookshop.ru/19258>.- ЭБС "IPRbooks" (дата обращения: 21.06.2016).
5. Гринченко Н.Н., Конкин Ю.В. Разработка моделей информационных систем на языке UML: учеб. пособие / РГРТУ. - Рязань, 2015. - 48с.
6. Конкин Ю.В. . Объектно-ориентированное программирование. Методические указания к лабораторной работе. Часть 8. № 4867. РГРТУ, Рязань, 2014.
7. Конкин Ю.В. . Объектно-ориентированное программирование. Методические указания к лабораторной работе. Часть 9. № 4905. РГРТУ, Рязань, 2015.

### **Дополнительная учебная литература**

1. Э. Фримен, К. Сиерра, Б. Бейтс Паттерны проектирования. – Спб.: «Питер», 2016. - 656с.
2. Аникеев С.В. Разработка приложений баз данных в Delphi : самоучитель. - М. : Диалог-МИФИ, 2013. - 160с.
3. Б. Маклафлин, Г. Поллайс, Д. Уэст Объектно-ориентированный анализ и проектирование. – Спб.: «Питер», 2014. - 285с.
4. М. Вайсфельд Объектно-ориентированное мышление. – Спб.: «Питер», 2014. - 340с.

## **8 ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

Обучающимся предоставлена возможность индивидуального доступа к следующим электронно-библиотечным системам:

Электронно-библиотечная система «IPRbooks», режим доступа – с любого компьютера

## **9 Методические указания для обучающихся по освоению дисциплины**

### **Рекомендации по планированию и организации времени, необходимого для изучения дисциплины**

Рекомендуется следующим образом организовать время, необходимое для изучения дисциплины:

Изучение конспекта лекции в тот же день, после лекции (10 – 15 минут).

Изучение конспекта лекции за день перед следующей лекцией (10 – 15 минут).

Изучение теоретического материала по учебнику и конспекту (1 час в неделю).

### **Описание последовательности действий студента («сценарий изучения дисциплины»)**

При изучении дисциплины рекомендуется самостоятельно изучать материал, который еще не прочитан на лекции и не применялся на лабораторном занятии. Тогда лекция будет гораздо понятнее. Но легче при изучении курса следовать изложению материала на лекции. Для понимания материала и качественного его усвоения рекомендуется следующая последовательность действий.

1. После прослушивания лекции и окончания учебных занятий, при подготовке к занятиям следующего дня, нужно сначала просмотреть и обдумать текст лекции, прослушанной сегодня (10 – 15 минут).

2. При подготовке к лекции следующего дня нужно просмотреть текст предыдущей лекции, подумать о том, какой может быть тема следующей лекции (10 – 15 минут).

В течение недели выбрать время (1 час) для работы с литературой.

### **Рекомендации по работе с литературой**

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта изучаются и книги по теории принятия решений. Литературу по дисциплине необходимо читать только в бумажном (не в электронном) виде. Полезно использовать несколько учебников и пособий по дисциплине. Рекомендуется после изучения очередного параграфа ответить на несколько вопросов по данной теме. Кроме того, полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): «о чем этот параграф?», «какие новые понятия введены, каков их смысл?».

## **10 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем**

При проведении занятий по дисциплине используются следующие информационные технологии:

– удаленные информационные коммуникации между студентами и преподавателем, ведущим лекционные и практические занятия, посредством информационной образовательной среды ФГБОУ ВО «РГРТУ», позволяющие осуществлять оперативный контроль графика выполнения и содержания образовательного процесса, решение организационных вопросов, консультирование;

– доступ к современным профессиональным базам данных (в том числе международным реферативным базам данных научных изданий) и информационным справочным системам.

**Перечень лицензионного программного обеспечения:**

- 1) Операционная система Windows XP Professional (лицензия Microsoft DreamSpark Membership ID 700102019);
- 2) Open Office (лицензия Apache License, Version 2.0);
- 3) Система программирования Lazarus.
- 4) Система программирования Visual Studio.

**10 Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

- 1) Интерактивная доска.
- 2) Мультимедиа-проектор.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Рязанский государственный радиотехнический университет им. В.Ф. Уткина»

КАФЕДРА ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ**

по дисциплине

**Б1.В.ДВ.05.01 «Объектно-ориентированное программирование»**

Направление подготовки  
38.03.05 – «Бизнес-информатика»

Направленность (профиль) подготовки  
«Бизнес-информатика»

Уровень подготовки - бакалавриат

Квалификация выпускника – бакалавр

Форма обучения – заочная

Рязань 2020 г

## 1 ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (практических заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности профессиональных компетенций.

Контроль знаний обучающихся проводится в форме промежуточной аттестации.

Промежуточная аттестация проводится в форме экзамена. Форма проведения экзамена - тестирование, письменный опрос по теоретическим вопросам и выполнение практических заданий.

## 2 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- 1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- 2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- 3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

### Уровень освоения компетенций, формируемых дисциплиной:

#### Описание критериев и шкалы оценивания тестирования:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 70 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 50 до 69%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 49%

#### Описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
0 баллов	выставляется студенту, который не смог ответить на вопрос

**Описание критериев и шкалы оценивания практического задания:**

<b>Шкала оценивания</b>	<b>Критерий</b>
3 балла (эталонный уровень)	Задача решена верно
2 балла (продвинутый уровень)	Задача решена верно, но имеются неточности в логике решения
1 балл (пороговый уровень)	Задача решена верно, с дополнительными наводящими вопросами преподавателя
0 баллов	Задача не решена

**На промежуточную аттестацию (экзамен, зачет) выносятся тест, два теоретических вопроса и 2 задачи.** Максимально студент может набрать 15 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно».

**Оценка «отлично»** выставляется студенту, который набрал в сумме 15 баллов (выполнил все задания на эталонном уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

**Оценка «хорошо»** выставляется студенту, который набрал в сумме от 10 до 14 баллов при условии выполнения всех заданий на уровне не ниже продвинутого. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

**Оценка «удовлетворительно»** выставляется студенту, который набрал в сумме от 5 до 9 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

**Оценка «неудовлетворительно»** выставляется студенту, который набрал в сумме менее 5 баллов или не выполнил всех предусмотренных в течение семестра практических заданий.

Промежуточная аттестация в форме зачета переводится в форму «зачтено», «незачтено»

**Оценка «зачтено»** выставляется студенту, который набрал в сумме более 5 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

**Оценка «незачтено»** выставляется студенту, который набрал в сумме менее 5 баллов или не выполнил всех предусмотренных в течение семестра практических заданий.

### 3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

<b>Контролируемые разделы (темы) дисциплины</b>	<b>Код контролируемой компетенции (или её части)</b>	<b>Вид, метод, форма оценочного мероприятия</b>
Теоретические основы объектно-ориентированного программирования	ПК-3	Экзамен
Принципы создания приложений Windows	ПК-3	Экзамен
Особенности объектной модели языка Object Pascal	ПК-3	Экзамен
Основы разработки приложений в управляемых вычислительных средах	ПК-3	Экзамен

## 4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

### 4.1. Промежуточная аттестация в форме экзамена

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ПК-3	выбор рациональных информационных систем и информационно-коммуникативных технологий решения для управления бизнесом

#### Типовые тестовые вопросы:

1. Основные принципы объектно-ориентированного программирования включают в себя:

+ инкапсуляцию, наследование, полиморфизм;  
виртуализацию, полиморфизм, наследование;  
абстрагирование, виртуализацию, наследование;  
наследование, виртуализацию, абстрагирование.

2. Свойство системы, позволяющее объединить данные и методы, работающие с ними, в одном классе, а также организовывать к ним отдельный доступ для различных частей программы называется:

виртуализацией;  
полиморфизмом;  
+инкапсуляцией;  
наследованием.

3. Свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью называется:

инкапсуляцией;  
виртуализацией;  
полиморфизмом;  
+наследованием.

4. Свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта называется:

инкапсуляцией;  
+полиморфизмом;  
виртуализацией;  
наследованием.

5. Класс - это:

+тип данных, включающий описание свойств и функций объектов;  
множество объектов с общим состоянием и поведением;  
тип данных, включающий описание функций с возможностью их повторного использования;

множество объектов, имеющих общие свойства и функции.

6. Объект - это:

абстрактный тип данных, включающий состояние и поведение;  
набор данных и функций работы с ними;  
+экземпляр класса;  
тип данных, реализующий функции класса и хранящий его состояние.

7. Скрытая часть класса описывается:

по умолчанию до первого спецификатора;  
после спецификатора public;  
после спецификатора protected;  
+после спецификатора private.

8. В примере:

```
TStudent = class
  private
    name : string;
  public
    function GetName;
    procedure SetName(aName : string);
end;
```

описание класса TStudent включает в себя:  
защищенную часть с данными и общую часть с методами;  
+скрытую часть с данными и общую часть с методами;  
общую часть с методами и данными;  
защищенную часть с методами и данными.

9. Защищенные элементы класса описываются:

после спецификатора public;  
+ после спецификатора protected;  
после спецификатора private;  
по умолчанию до первого спецификатора.

10. Общедоступные элементы класса описываются:

после спецификатора protected;  
после спецификатора private;  
+после спецификатора public;  
по умолчанию до первого спецификатора.

11. К общедоступным свойствам класса (public) разрешен доступ:

только из методов данного класса и классов потомков;  
только из методов данного класса;  
только из методов данного класса и классов предков;  
+из всех методов программы.

12. К защищенным свойствам класса (protected) разрешен доступ:

из всех методов программы;  
только из методов данного класса;  
только из методов данного класса и класса предка;  
+только из методов данного класса и классов производных от данного.

13. К закрытым свойствам класса (private) разрешен доступ:

только из методов данного класса, классов потомков и классов предков;  
+только из методов данного класса  
из всех методов программы  
только из методов данного класса и классов потомков;

14. Функциям базового класса разрешен доступ:

+ко всем свойствам и методам базового класса;

только к защищенным свойствам и методам базового класса;  
только к закрытым свойствам и методам базового класса;  
только к защищенным и общим свойствам и методам базового класса.

15. Функциям класса потомка разрешен доступ:  
ко всем свойствам и методам базового класса;  
+к защищенным и общим свойствам и методам базового класса;  
только к защищенным свойствам и методам базового класса;  
к закрытым и общим свойствам и методам базового класса.

16. В примере:

```
TStudent = class
  public
    aName string;
end;
```

для описания конструктора в классе TStudent необходимо добавить выражение:

```
procedure Student(aName : string);
Student(aName : string);
+constructor Create(aName : string);
Createt(aName : string);
```

17. Конструктор класса используется для:

+инициализации объектов класса;  
инициализации класса для наследования;  
инициализации общих свойств класса;  
начального присвоения значений переменным класса.

18. Деструктор класса используется для:

удаления объектов базового класса;  
+освобождение памяти, выделяемой при инициализации объектов класса;  
освобождение памяти, выделяемой для динамических переменных класса;  
удаление динамических переменных и указателей класса.

19. В примере:

```
TPerson = class
  public
    name string;
end;
TStudent = class
  public
    group integer;
    procedure show;
end;
```

для того, чтобы класс Student стал наследуемым от класса Person необходимо:

+1: изменить заголовок класса Student: TStudent = class(TPerson)

2: изменить заголовок класса Person:

```
TPerson = class(TStudent)
```

3: добавить конструктор класса Person с описанием:

```
Constructor Create (name : string);
```

4: добавить конструктор класса Student с описанием:

```
Constructor Create(name : string; gr : integer);
```

20. В примере:

```
TPerson = class
  private
    string name;
  public:
    procedure Person(n : string);
    function Get : string;
    procedure Print; virtual;
end;
```

```
TStudent = class(TPerson)
  private
    group integer;
  public:
    procedure Set(g : integer);
    procedure Print; override;
    function Get1 : integer;
end;
```

полиморфными являются следующие методы класса TStudent:

- 1: только метод Set;
- 2: методы Set и Print;
- +3: только метод Print;
- 4: функции Get1, Set и Print.

21. Для обработки исключительных ситуаций используется оператор:

```
+try...except;
if...then...else;
while...do;
exit.
```

22. Виртуальным называется метод , который:

объявлен в абстрактном классе;

не имеет реализации в классе;

может быть переопределена в текущем классе и классах потомков;

+: может быть переопределен в классах потомков так, что конкретная его реализация определяется в ходе выполнения программы.

23. Абстрактным классом называется класс:

1: содержащий хотя бы один виртуальный метод;

+2: содержащий хотя бы один метод, не имеющий реализации;

3: не имеющий методов;

4: не имеющий объектов.

24. Метод класса определяется для:

1: абстрактного класса;

+2: класса, для которого не создается объект;

3: базового класса в иерархии классов;

4: класса с виртуальными методами.

25. Абстрактный класс предназначен для:

создания объектов базового класса;  
создания одного экземпляра базового класса;  
+определения реализации части функций в классах потомках;  
определения общих свойств для всех классов проекта.

26. Абстрактный класс не может быть использован для:  
определения реализации части функций в классах потомках;  
определения данного класса как базового;  
инициализации общих свойств базового класса;  
+создания объектов.

27. В примере:

```
TBase = class
  private
    x integer;
  public
    procedure Set;
    procedure Print;
}
```

для того, чтобы сделать класс TBase абстрактным, метод Print должен описываться следующим образом:

```
procedure Print; override;
procedure Print; dynamic;
+procedure Print; virtual; abstract;
procedure Print; virtual;
```

28. Делегирование методов это:

вызов для обработки объекта данного класса метода базового класса;  
+ вызов для объекта данного класса метода другого класса с помощью указателя;  
вызов для объекта данного класса метода производного класса;  
вызов для объекта данного класса метода любого абстрактного класса.

29. Значением переменной типа метакласса является:

+наименование класса для заданной иерархии классов;  
ссылка на любой другой класс;  
указатели на абстрактные классы;  
указатели на контейнерные классы.

30. Переменная типа класса содержит:

объект данного класса;  
экземпляр данного класса;  
+указатель на объект данного класса;  
значения полей данного класса.

#### **Типовые практические задания:**

##### **Задание 1**

Подготовить и выполнить программу на языке Object Pascal, в которой формируется и выводится список студентов и преподавателей с использованием классов *TPerson*, *TStud*, *TProf*.

```
// базовый класс
```

```
TPerson=class
```

```
  fname:string; { имя }
```

```

    constructor Create(name:string);
    function info: string; virtual;
end;
// производный от базового TPerson
TStud=class(TPerson)
    fgr:integer; { номер группы }
    constructor Create(name:string;gr:integer);
    function info: string; override;
end;
// производный от базового TPerson
TProf=class(TPerson)
    fdep:string; { название кафедры }
    constructor Create(name:string;dep:string);
    function info: string; override;
end;

```

#### ***Критерии выполнения задания 1***

Задание считается выполненным, если: обучающийся на основе описания заданной иерархии классов предметной области разработал алгоритмы формирования объектов классов и использовал их для решения задачи.

#### ***Задание 2***

Подготовить и выполнить программу на языке Object Pascal, выводящую сообщение о названии типа созданного класса и названии его непосредственного предка.

#### ***Критерии выполнения задания 2***

Задание считается выполненным, если: обучающийся правильно использовал методы класса TObject для решения задачи.

#### ***Задание 3***

Подготовить и выполнить программу, выводящую сообщение о характеристиках любого класса – название его типа и название типа его непосредственного предка.

#### ***Критерии выполнения задания 3***

Задание считается выполненным, если обучающийся правильно использовал методы класса и указатели на классы (метаклассы) для решения задачи.

#### ***Задание 4***

Подготовить и выполнить программу построения графика функции  $Y=F(X)$ , которая задается следующим образом:

$$\begin{aligned}
 0 &\leq X < 50 & Y &= 0, \\
 50 &\leq X < 100 & Y &= X - 50, \\
 100 &\leq X < 150 & Y &= 50 \\
 150 &\leq X < 200 & Y &= -(X - 200), \\
 200 &\leq X < 250 & Y &= 0.
 \end{aligned}$$

#### ***Критерии выполнения задания 4***

Задание считается выполненным, если обучающийся разработал алгоритм табулирования функции и вывода графика на экран.

#### ***Задание 5***

Разработать программу на языке Object Pascal в системе программирования Delphi, в которой выполняется открытие и чтение информации из текстового файла. Доступ к файлу должен контролироваться обработчиками исключительных ситуаций.

#### ***Критерии выполнения задания 5***

Задание считается выполненным, если обучающийся использовал в программе операторы обрабатывающей и завершающей конструкций обработки исключений.

#### ***Задание 6***

Разработать программу на языке Object Pascal в системе программирования Delphi, в которой выполняется создание и вывод на экран списка студентов на основе контейнерного класса.

#### ***Критерии выполнения задания 6***

Задание считается выполненным, если обучающийся организовал обработку списка на основе процедур, представленных методами контейнерного класса.

#### **Типовые теоретические вопросы:**

1. Основные этапы развития программирования.
2. Структурная декомпозиция
3. Модульное программирование.
4. Отличительные особенности объектно-ориентированного подхода в программировании.
5. Архитектура объектно-ориентированных программ.
6. Основные принципы объектно-ориентированного программирования.
7. Этапы разработки программного обеспечения при ООП.
8. Понятие объектной декомпозиции. Пример.
9. Понятие объекта. Объекты и сообщения. Состояние и поведение объектов.
10. Понятие объекта. Операции над объектами.
11. Понятие класса. Структура и организация определения класса.
12. Определение классов и объектов на языке Object Pascal.
13. Конструирование и уничтожение объектов в языке Object Pascal.
14. Особенности работы с объектами в языке Object Pascal.
15. Средства разработки классов. Наследование. Инициализация объектов производных классов. Управление доступом к элементам класса в связи с наследованием.
16. Полиморфизм как средство разработки классов.
17. Понятие о статическом и динамическом связывании.
18. Простой полиморфизм. Пример.
19. Сложный полиморфизм и создание полиморфных объектов. Пример.
20. Виртуальные методы как средство реализации сложного полиморфизма.
21. Композиция и наполнение как средства разработки классов.
22. Обработка исключительных ситуаций в программах на языке ООП.
23. Метаклассы.
24. Делегирование методов как средство разработки классов.
25. Контейнерные классы.
26. Параметризованные классы.
27. Свойства в языке Object Pascal. Простые свойства.
28. Свойства в языке Object Pascal. Свойства массивы.
29. Свойства в языке Object Pascal. Индексные свойства.
30. Делегирование методов и процедурные свойства.
31. Система сообщений Windows.